

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

SYSTEM AND METHODS FOR HIGH RATE HARDWARE-
ACCELERATED NETWORK PROTOCOL PROCESSING

Wang et al.

Appl. No.: Unknown Atty Docket: ISTOR.012A

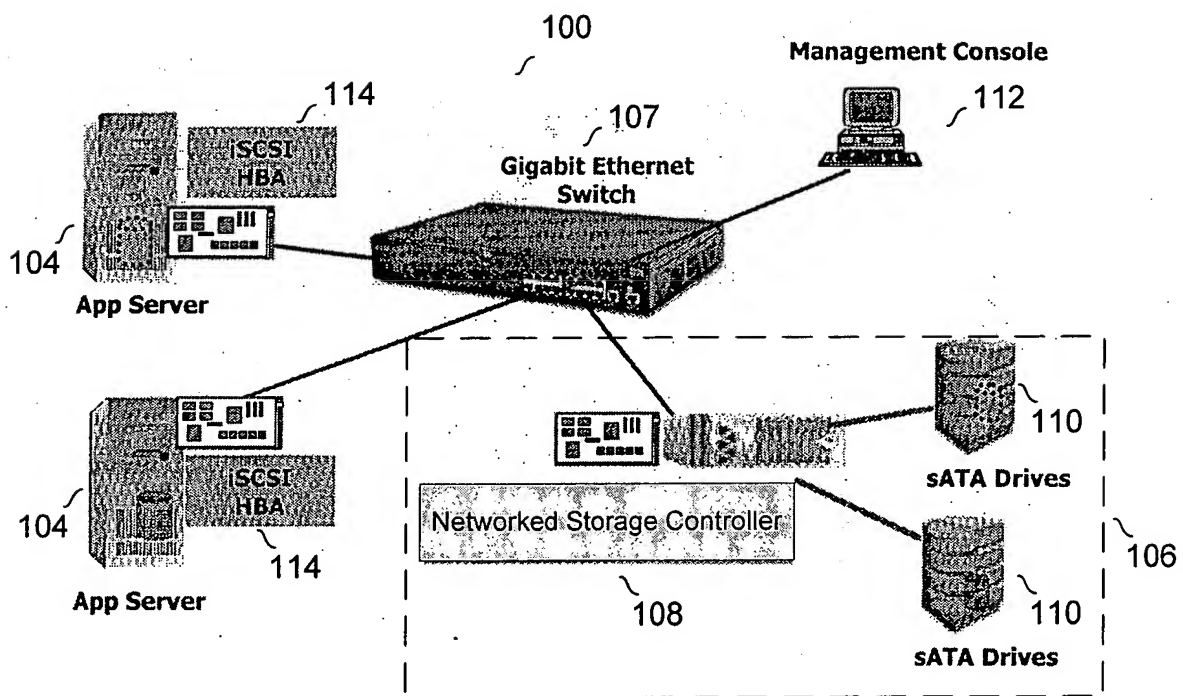


FIGURE 1

SYSTEM AND METHODS FOR HIGH RATE HARDWARE-
ACCELERATED NETWORK PROTOCOL PROCESSING

Wang et al.

Appl. No.: Unknown

Atty Docket: ISTOR.012A

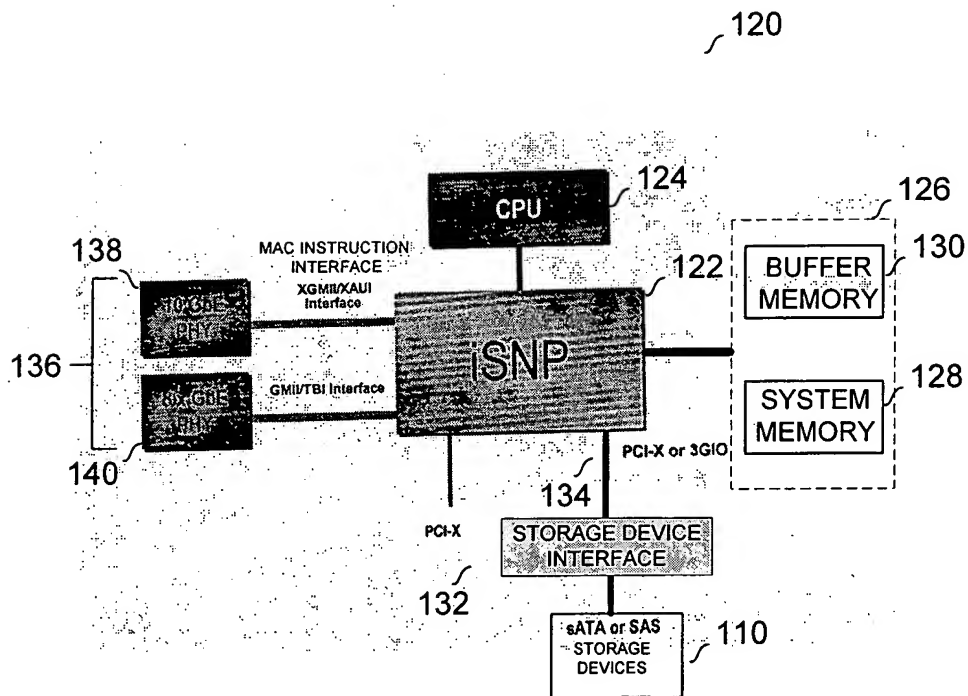


FIGURE 2

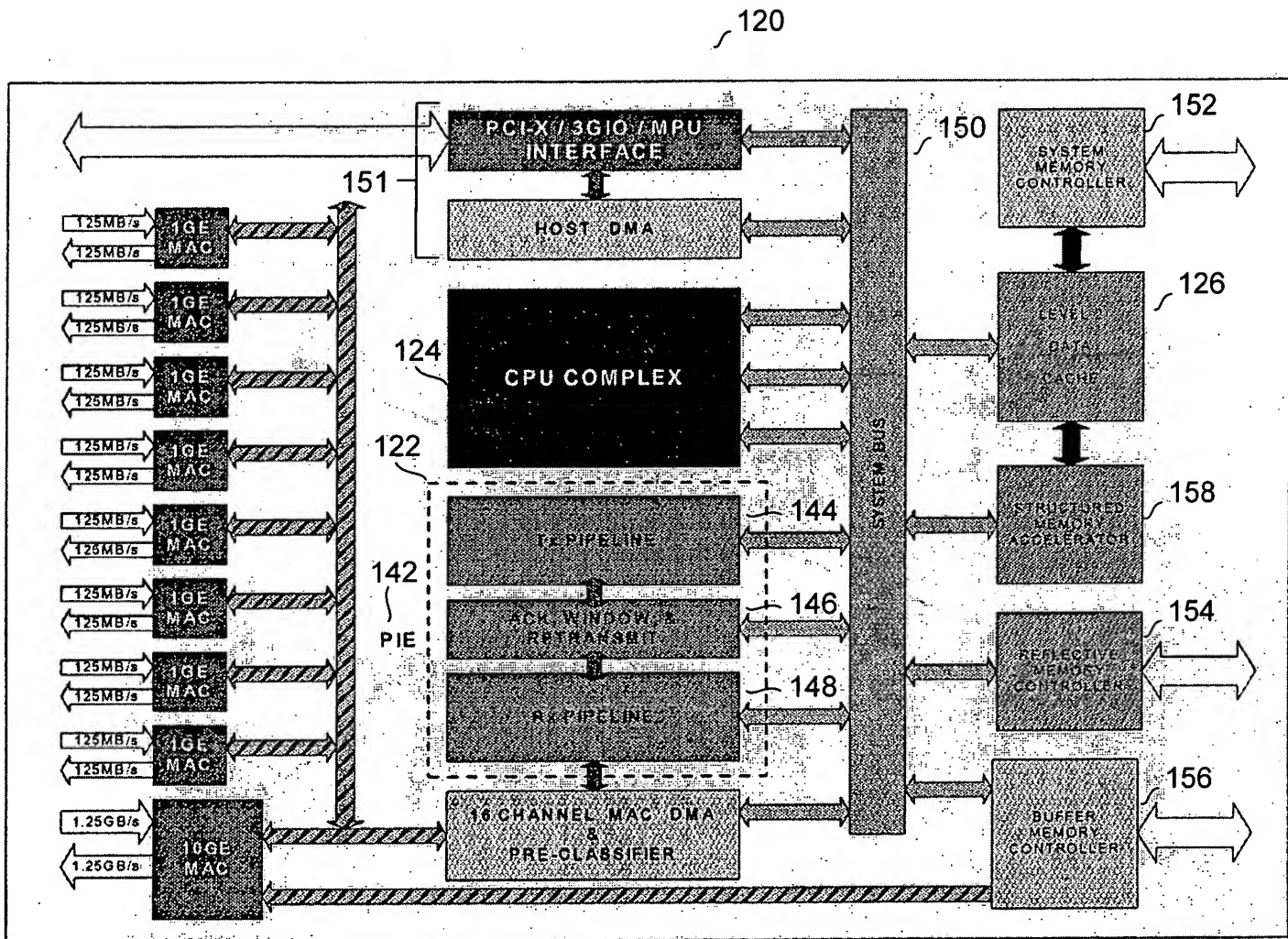


FIGURE 3A

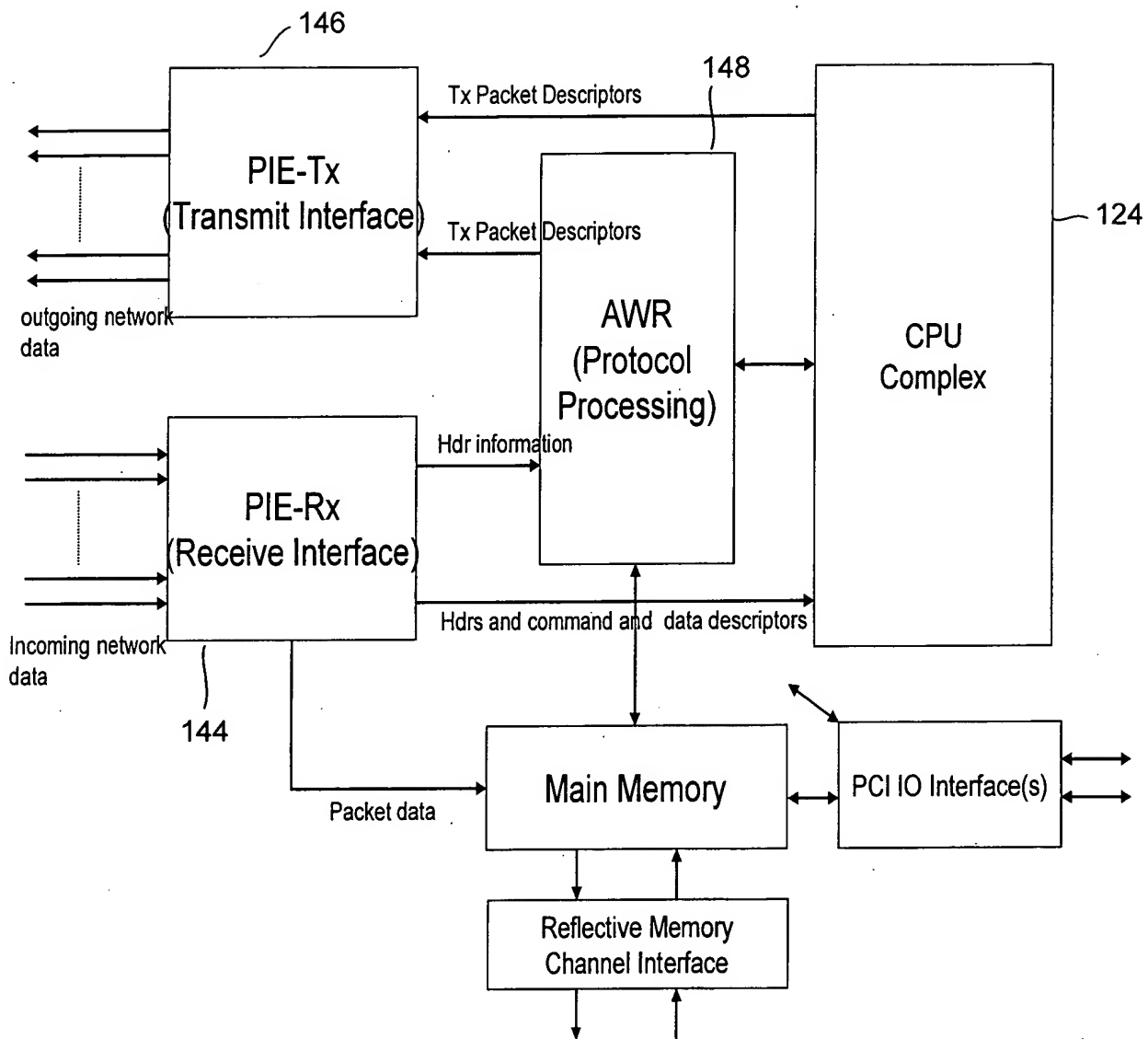


FIGURE 3B

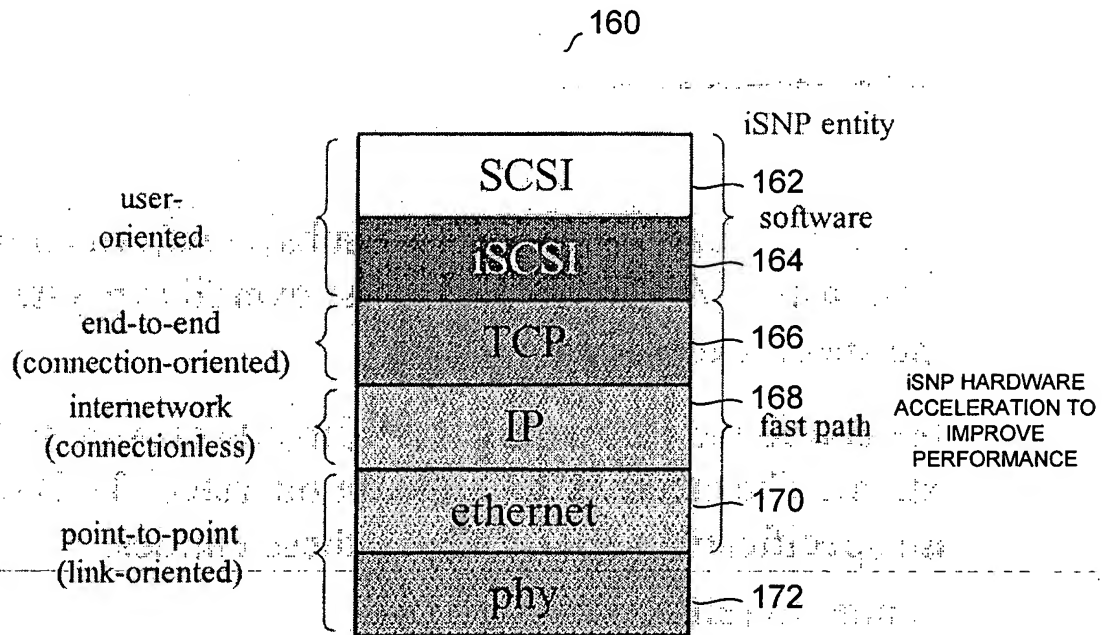


FIGURE 4A

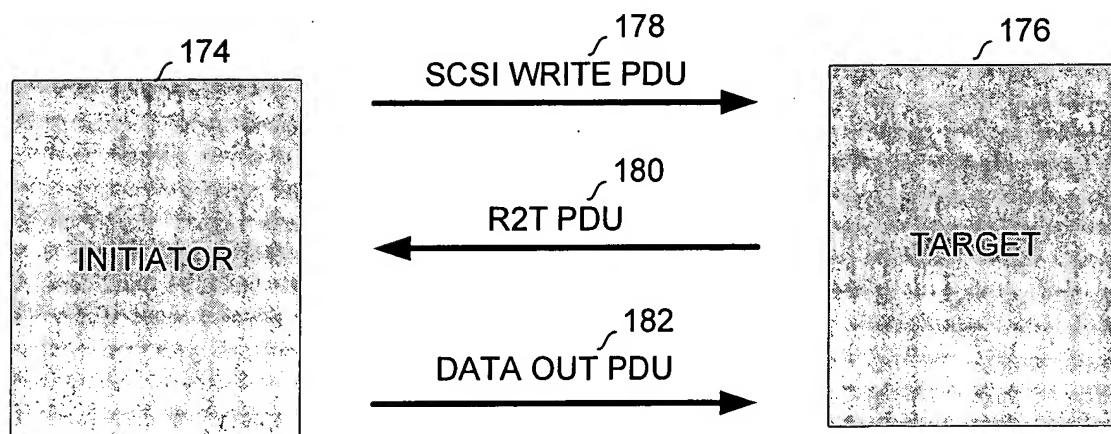


FIGURE 4B

Appl. No.: Unknown Atty Docket: ISTOR.012A

FIGURE 5AFIGURE 5B

SYSTEM AND METHODS FOR HIGH RATE HARDWARE-
ACCELERATED NETWORK PROTOCOL PROCESSING

Wang et al.

Appl. No.: Unknown Atty Docket: ISTOR.012A

word#	IP Packet Header Fields			
0	version[3:0] (header len in words, min 5=no options)	IHL[3:0] (header len in words, min 5=no options)	TOS[7:0] (type of service, specifies precedence, delay, throughput, and reliability parameters)	TLLEN[15:0] (total length including header, in octets)
1	ID[15:0] (with src addr, dst addr, and user protocol uniquely identifies the IP datagram)			flag[2:0] flag[1]=dont frag flag[0]=more frags fragment offset[12:0] (in 64-bit units)
2	TTL[7:0] (time to live, decremented at each hop, if 0 discard the datagram)		Protocol[7:0] 1=ICMP 6=TCP 17=UDP	Header checksum [15:0] (1's complement of the 1's complement sum of data interpreted 16 bits at a time, with end-around carry)
3	Source IP address[31:0] (needed to allow a variable number of bits to specify the network and station)			
4	Destination IP address[31:0] (as for source address, 224-239.x.x.x=multicast)			
:	Options (if any) plus padding (variable length depending on the number and type of options)			
:	Data (multiple of 8 bits in length)			

FIGURE 5C

ethernet frame field	pre-amble	SFD (start frame)	DA (dst addr)	SA (src addr)	type (ethernet II) or length of info (IEEE 802.3)	information	FCS (frame chk seq)	extension
#octets	7	1	6	6	2	46-1500	4	
notes					'h0800=IP 'h0806=ARP 'h86dd=IPv6 'h8100=VLAN (insert w/2-byte tag before type/length) <='h05dc=length (rfc1042, insert w/constant 0xaaaa03 000000 before type)	jumbo: max 9000 GE: min=512 for half duplex CSMA/CD <min: add pad bytes		special nondata symbols for half duplex CSMA/CD

FIGURE 5D

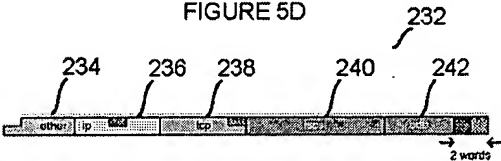


FIGURE 5E

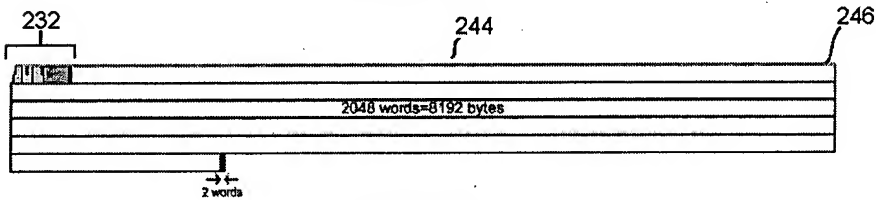


FIGURE 5F

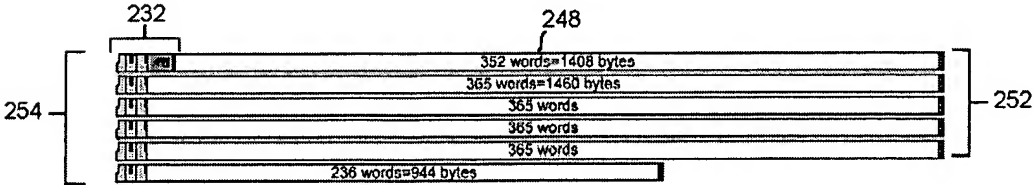
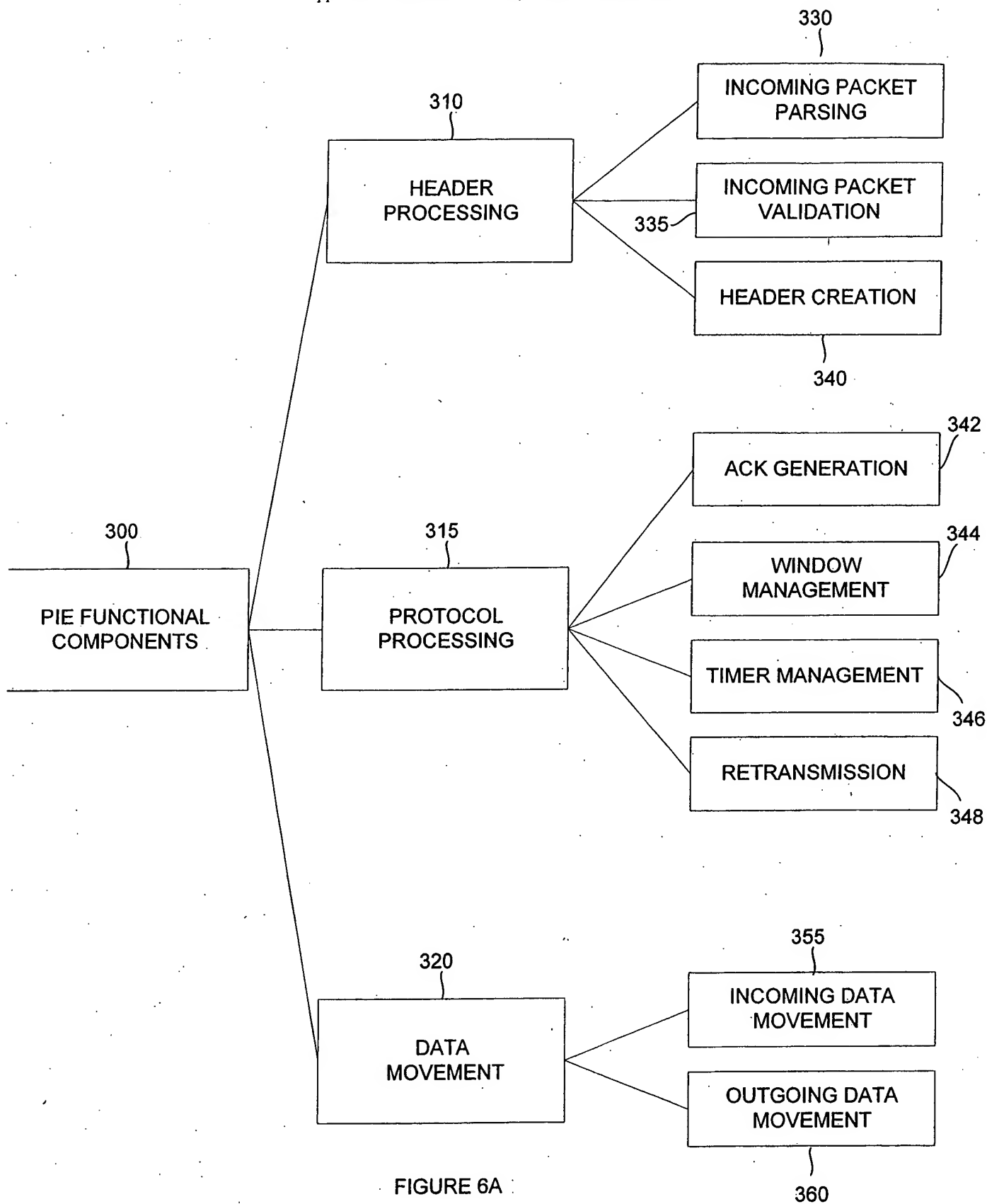


FIGURE 5G



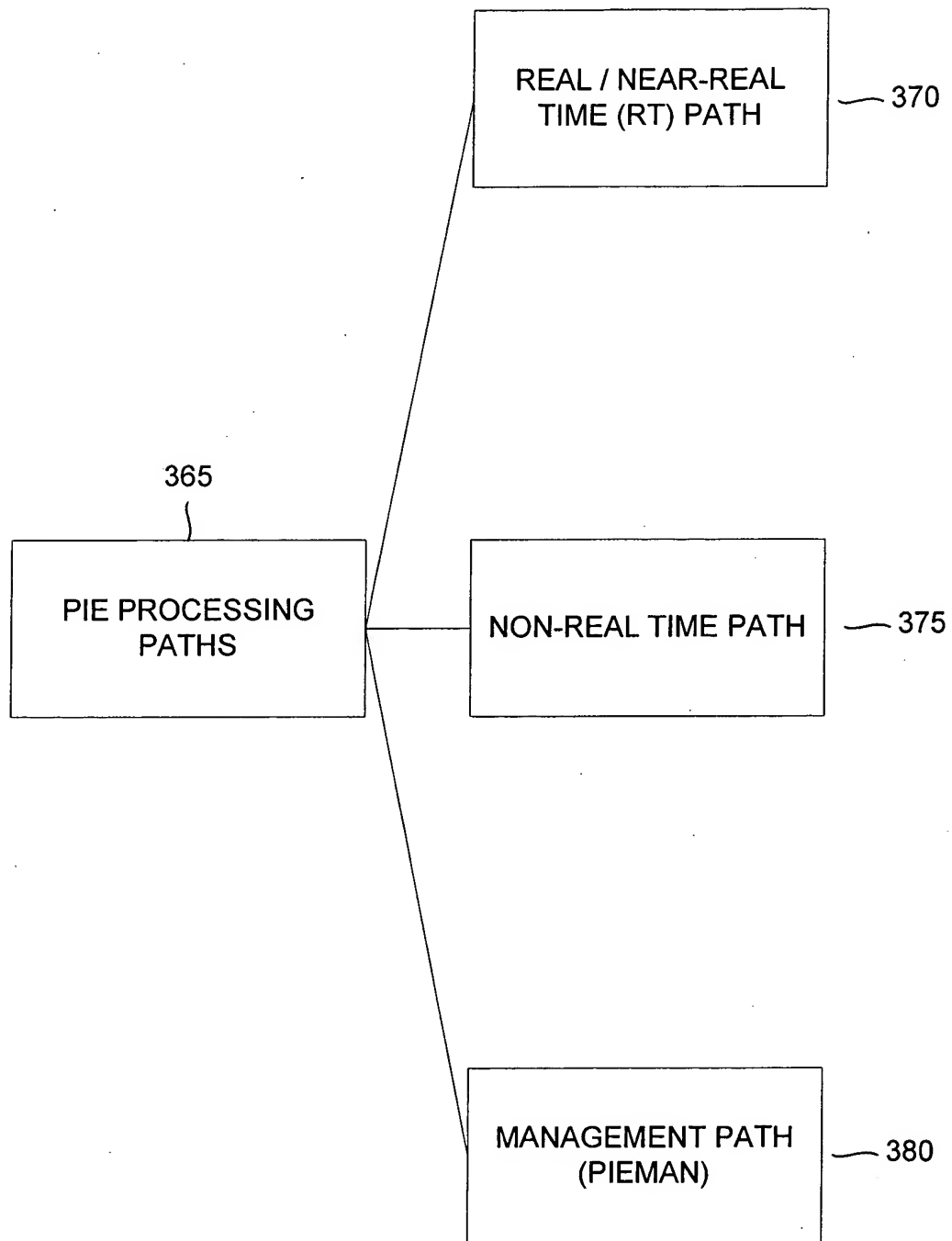


FIGURE 6B

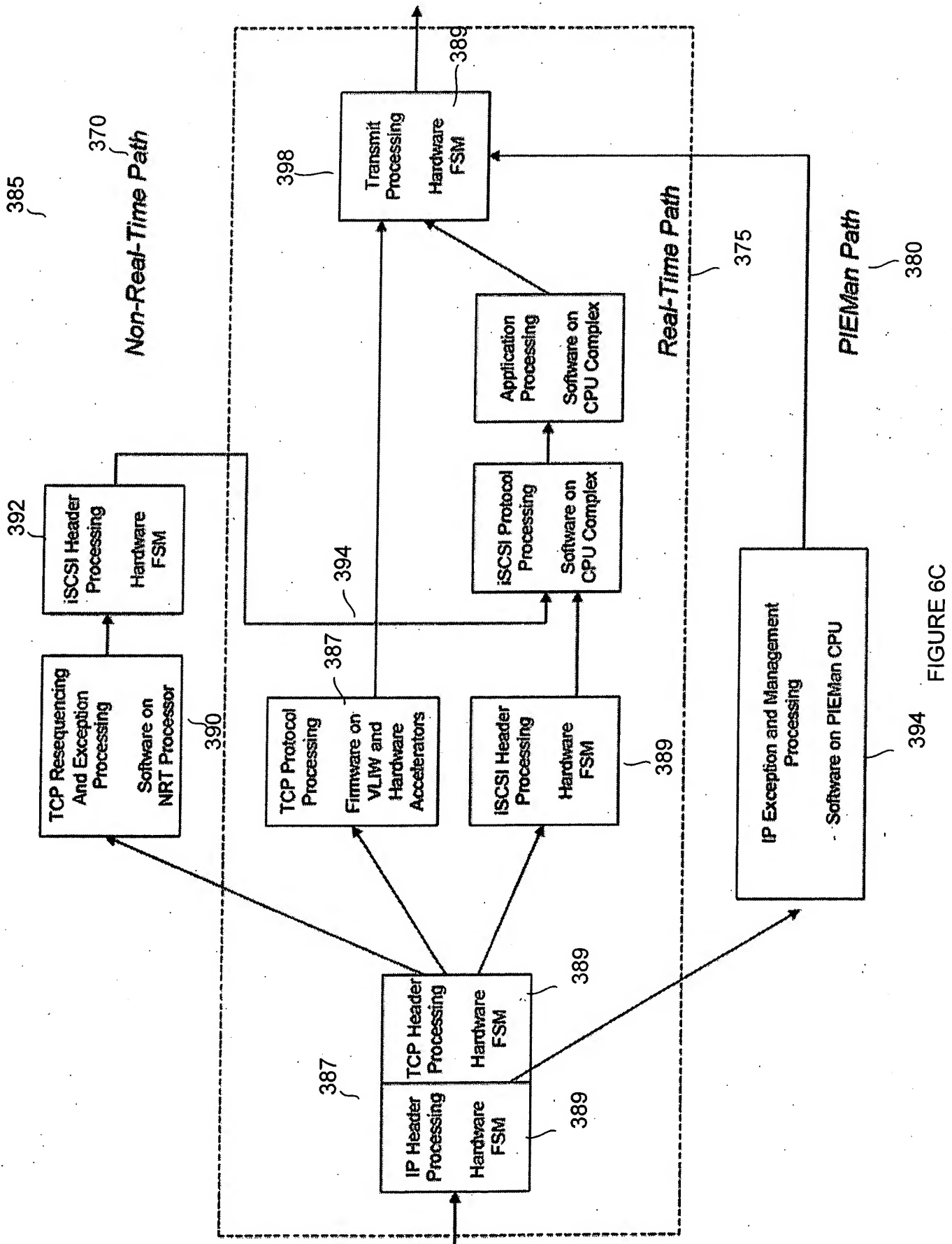
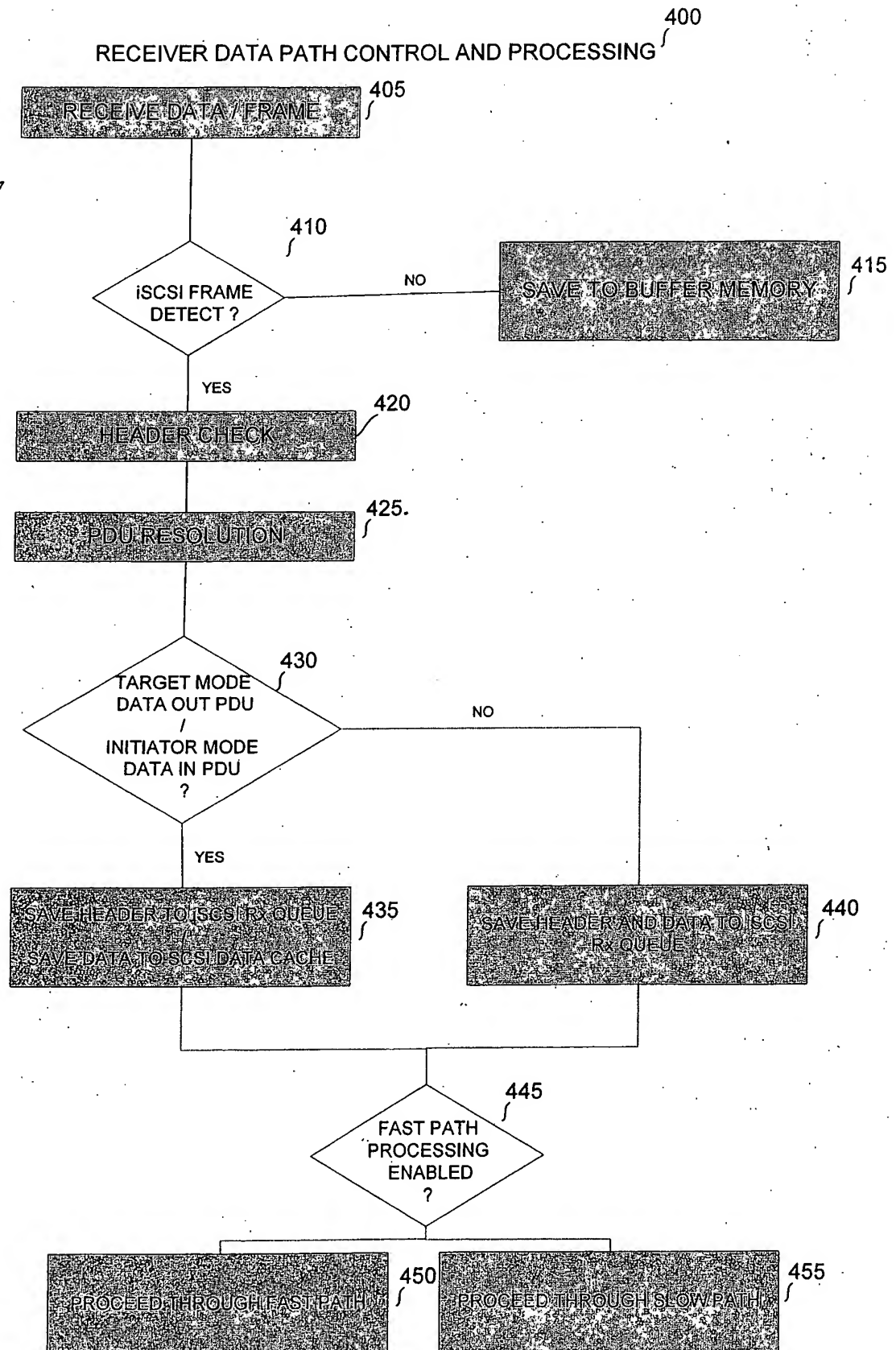


FIGURE 6C

FIGURE 7



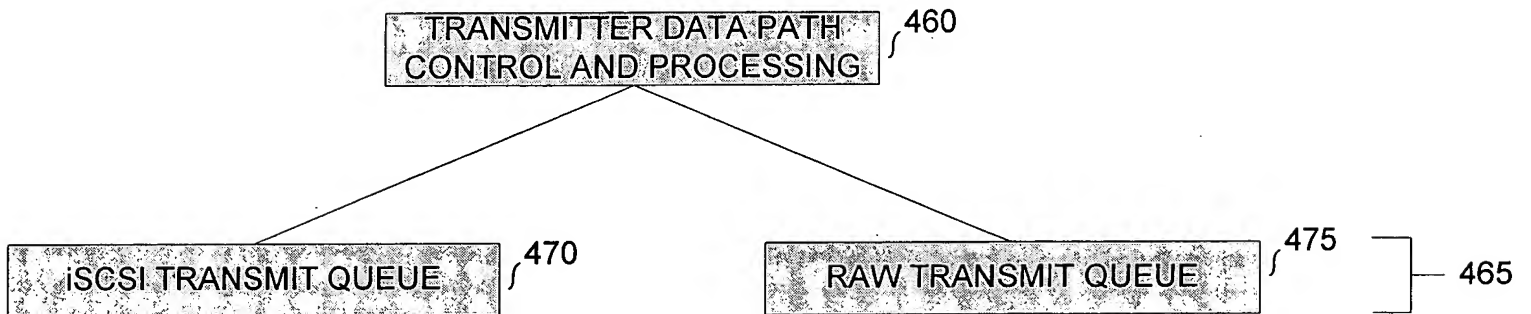


FIGURE 8A

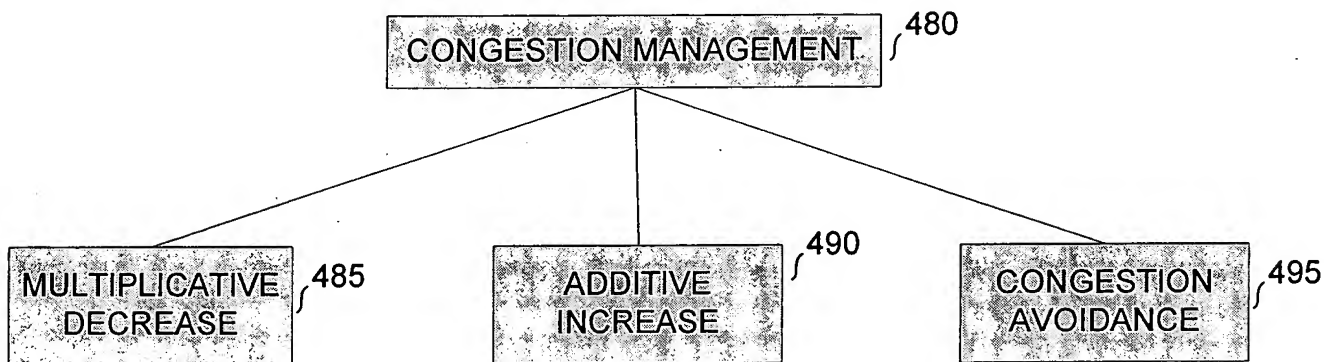
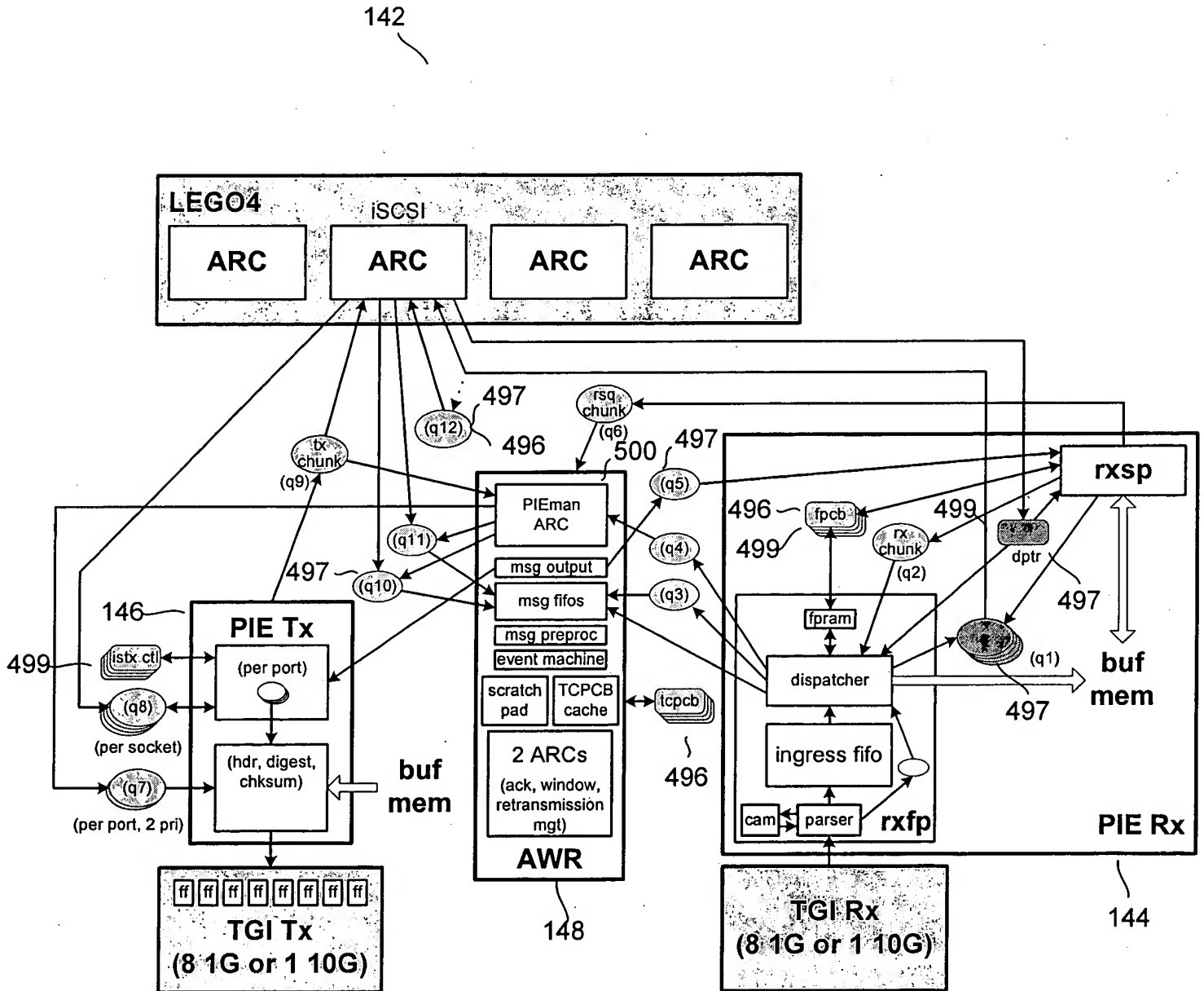


FIGURE 8B



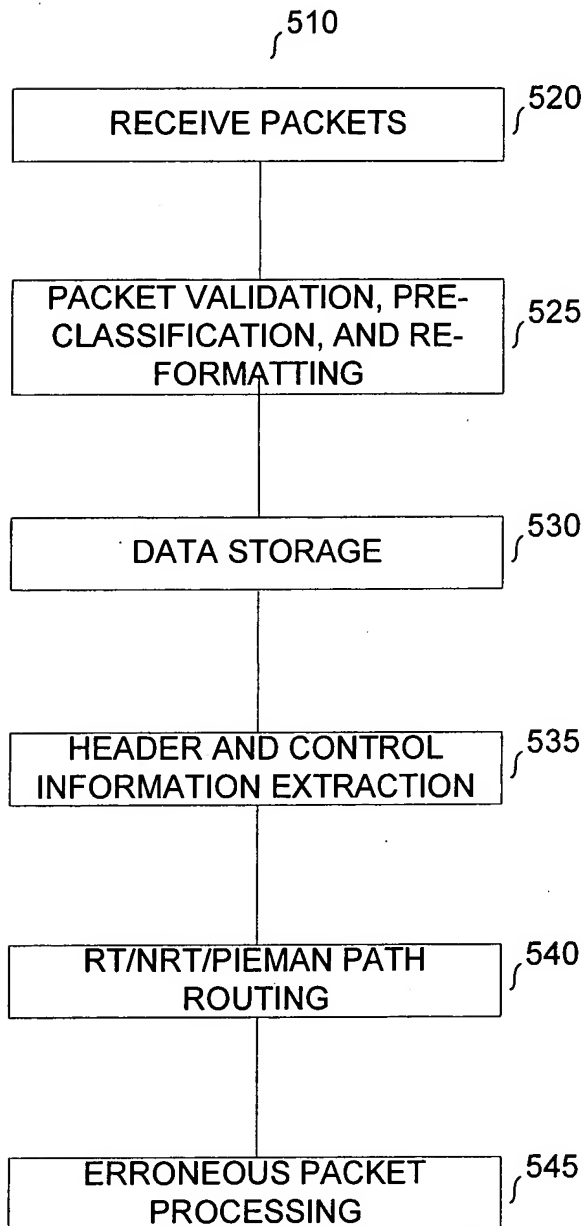


FIGURE 10

SYSTEM AND METHODS FOR HIGH RATE HARDWARE-
ACCELERATED NETWORK PROTOCOL PROCESSING

Wang et al.

Appl. No.: Unknown Atty Docket: ISTOR.012A

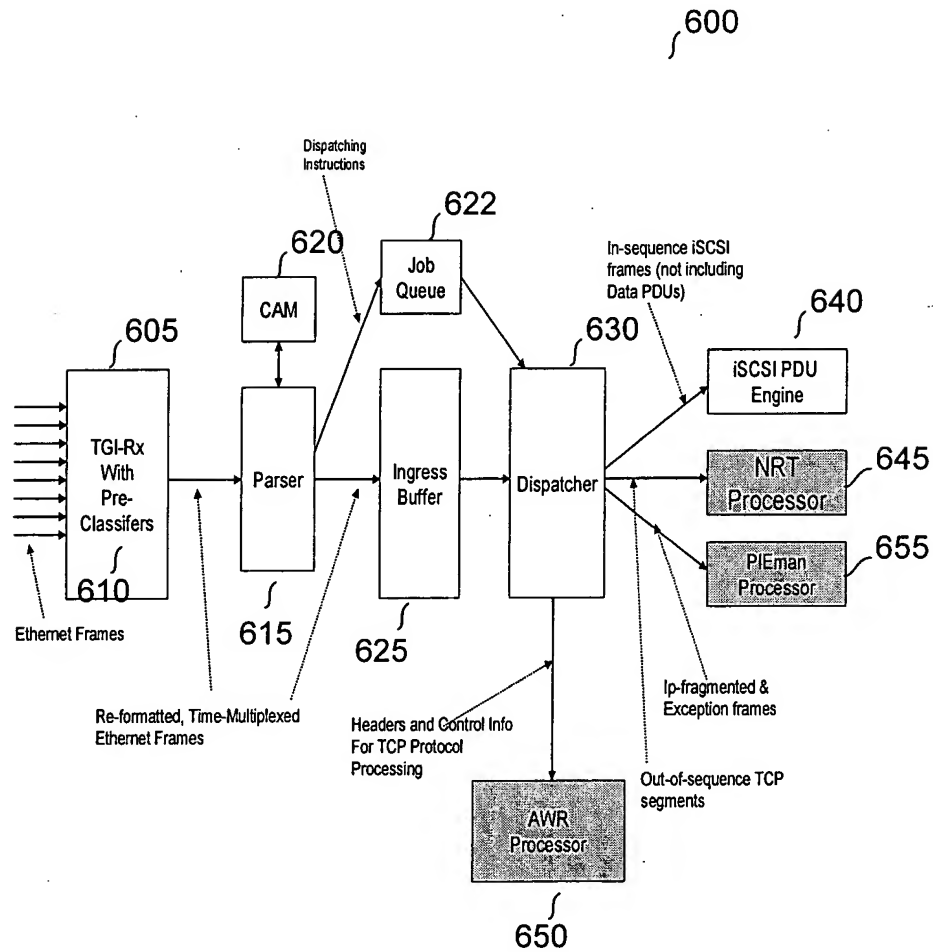


FIGURE 11

TGI field	Description
tag[3:0]	<p>Indicates the type of information in the dword, as preclassified by the TGI</p> <p>0: invalid (interframe) 1: E (ethernet header) 2: EV (ethernet header, VLAN present) 3: E8 (ethernet header, 802.3 rfc1042 format) 4: EV8 (ethernet header, 802.3 rfc1042 format, VLAN present) 5: I (IP) 6: IO (IP options) 7: IF (IP fragmented)</p> <p>8: T (TCP) 9: TO (TCP option) 10: U (UDP) 11: spare 12: S (iSCSI, i.e. TCP and DPORT or SPORT matches iSCSI) 13: O (other, e.g. not IP, not TCP or UDP) 14: G (good EOF, dword holds checksum, frame length) 15: B (bad EOF, assert early if checksum or other error detected)</p>
off[2:0]	Indicates the byte offset to tag boundary within dword, 0=left side

FIGURE 12A

B0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
raw non-vlan and vlan frames

ethernet-ii and ethernet-frames																			
da				sa				type				hlen							
hlen	hlen	frag	iprot				cksm	tsip				dip							
dport				sport				seqnum				acknum							
win				cksm				urc											
da				sa				vlan				vtag							
type				hlen				hlen	hlen	frag	iprot				cksm	sip			
spit				dp				sport				dport				seqnum			
ack				win				cksm				urc							
da				sa				vlan				vtag							
802.3 len: pat: ip																			
type				hlen				hlen				hlen							
frag				iprot				cksm	tsip				dp						
sport				dport				seqnum				acknum							
win				cksm				urc											
da				sa				802.3 u											
802.3 l				type				hlen				hlen	hlen	frag	iprot				
cksm				tsip				dp				sport				dport			
seqnum				acknum				win				cksm				urc			

B0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
tag,off formatted non-vlan and vlan frames

formatted non-vlan and vlan frames																	
e 0	da				sa				type		0						
i 0	hlen	len	id	frag	iprot	cksm	sip										
t 4	dip	sport		dport	seqnum		acknum										
t 8	hlen	win	cksm		urc												
ev 0	da				sa				type		vtag						
i 0	hlen	len	id	frag	iprot	cksm	sip										
t 4	dip	sport		dport	seqnum		acknum										
t 8	hlen	win	cksm		urc												
ev8 0	da				sa				type		vtag						
i 0	hlen	len	id	frag	iprot	cksm	sip										
t 4	dip	sport		dport	seqnum		acknum										
t 8	hlen	win	cksm		urc												
e8 0	da				sa				type		0						
i 0	hlen	len	id	frag	iprot	cksm	sip										
t 4	dip	sport		dport	seqnum		acknum										
t 8	hlen	win	cksm		urc												

FIGURE 12B

#bits	Field	Description
4	State	State machine state
2	ts_offst	Timestamp offset (0=none, 1=22B from start of TCP header, 2=23B, 3=24B)
1	t_left	TCP header starts in left half of dword
4	Reason	If nonzero, the reason from slow-path processing
1	Msw_parity	Job FIFO MS word parity
1	d_left	1 st dword stored (at next dword write qword to ingress fifo)
64	d_data	Stored dword
2	d_par	Stored dword parity
80		Total bits

FIGURE 13A

Slow-Path Reason codes (*=set by dispatcher):

- 0 : nop -- fastpath iSCSI
- 1: ARP frame
- 2: other non-IP (not ARP) frame
- 3: IP fragment (if not, fragment zero could be iSCSI)
- 4: TCP but not iSCSI or runt iSCSI (flen<0x38)
- 5: UDP frame
- 6: ICMP frame
- 7: other IP frame (not IP fragment, TCP, UDP, or ICMP)
- 8: iSCSI, IP fragment zero
- 9: iSCSI, no socket ID found in CAM
- a: iSCSI, unsupported option
- *b: iSCSI, fastpath disabled
- *c: iSCSI, out of sequence
- *d: iSCSI, bad data boundary

FIGURE 13B

CAM_LOAD I, data = Write CAM entry I with specified data; Set Valid bit;
CAM_READ I = Read data contained in CAM entry I;
CAM_INV I = Clear the valid bit for CAM entry I;
CAM_REQ P = Initiate CAM search with Key elements written in the
CAM-Key register for network port P;
CAM_RESULT P = Fetch result from CAM search for network port P;

FIGURE 14

#bits	Field	Description
2	TS offset	If nonzero, byte offset minus 1 from start of TCP options to timestamp field
4	TCP option length	Size in words of TCP options
4	Slow-path Reason Code	<p>If nonzero, packet takes slow path.</p> <p>Reason codes (*=set by dispatcher):</p> <p>0: nop, fastpath iSCSI</p> <p>1: ARP</p> <p>2: other non-IP, non-ARP frame</p> <p>3: IP fragment</p> <p>4: TCP (not iSCSI) or runt iSCSI (flen<0x38)</p> <p>5: UDP</p> <p>6: ICMP</p> <p>7: other IP (not IP fragment, TCP, UDP, or ICMP)</p> <p>8: iSCSI, IP fragment zero</p> <p>9: iSCSI, no socket ID</p> <p>a: iSCSI, unsupported option</p> <p>*b: iSCSI, fastpath disabled</p> <p>*c: iSCSI, out of sequence</p> <p>*d: iSCSI, bad data boundary</p>
1	ID valid	iSCSI socket ID valid
1	Init	Initiator mode
4	IP option length	Size in words of IP options
10	Socket ID	iSCSI socket ID
1	VLAN	16 th byte contains VLAN tag (IP frame)
1	802.3	802.3 rfc1042 coding was removed from ethernet header (IP frame)
14	frame length	Length of formatted frame in bytes
16	partial checksum	Checksum for UDP or partial TCP segment (info for PIEman)

FIGURE 15

#bits	field	description	notes	
58	job	jff output		
13	roffst	iff read offset	for random access of iff	
13	rckpt	iff read checkpoint (start of frame)	for calculating iff discard point (add flength)	
10	fctr	frame qword counter		
32	seq	TCP sequence number	if iSCSI (except flags)	for msgRxNotify and msgRxFrame, also need some job fields
32	ack	TCP acknowledgement number		
32	flgs	flags, TCP flags, TCP window size		
32	ts	TCP timestamp		
32	ets	TCP echo timestamp		
288	rcp0-rcp8	Rx chunk pointer 0 up to 8	depending on flength	for msgRxFrame
546		total (approx 69 bytes per port, total 552B)		

Figure 16 . Dispatcher Per-port Frame Context

FIGURE 16

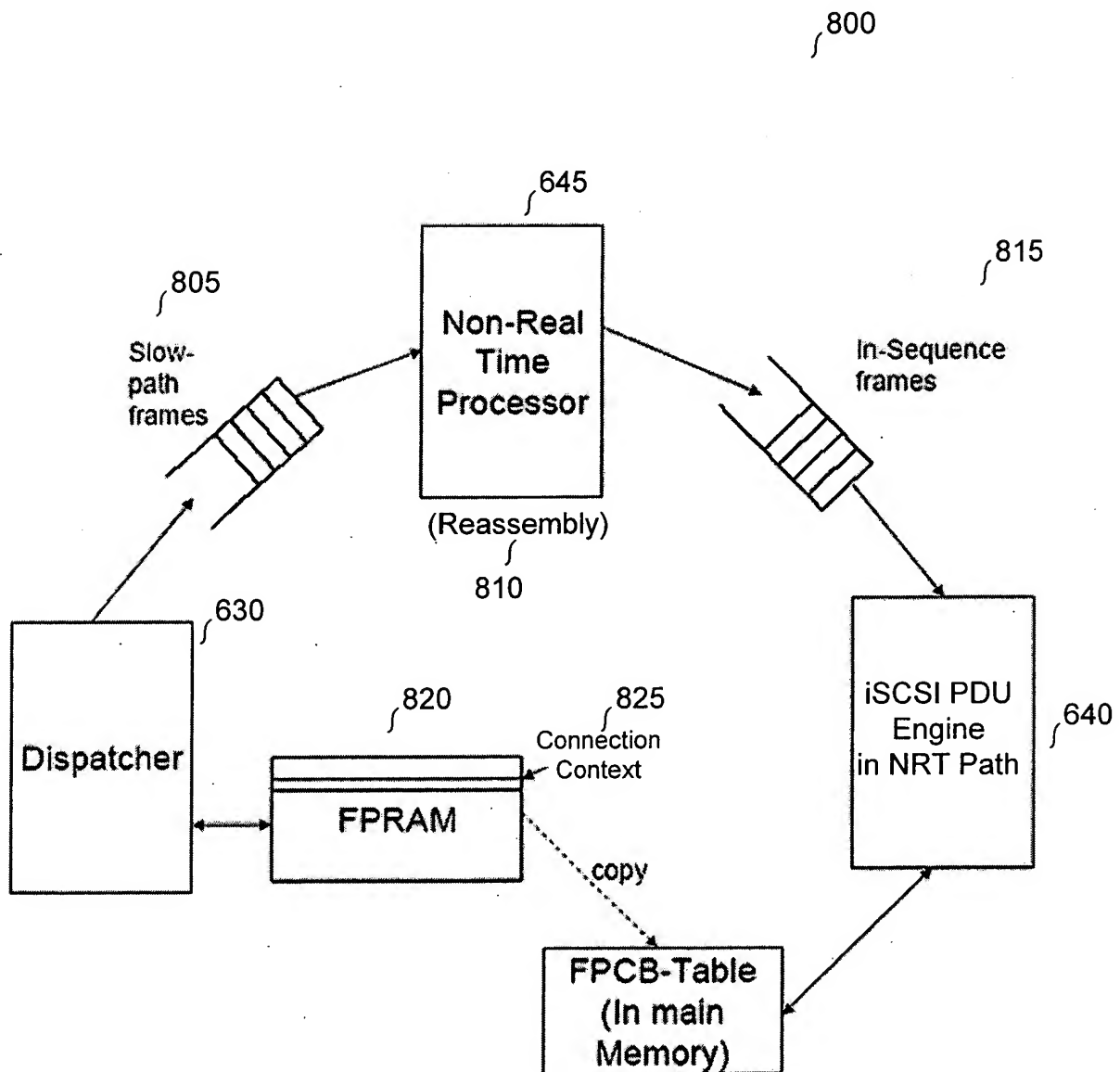


FIGURE 17

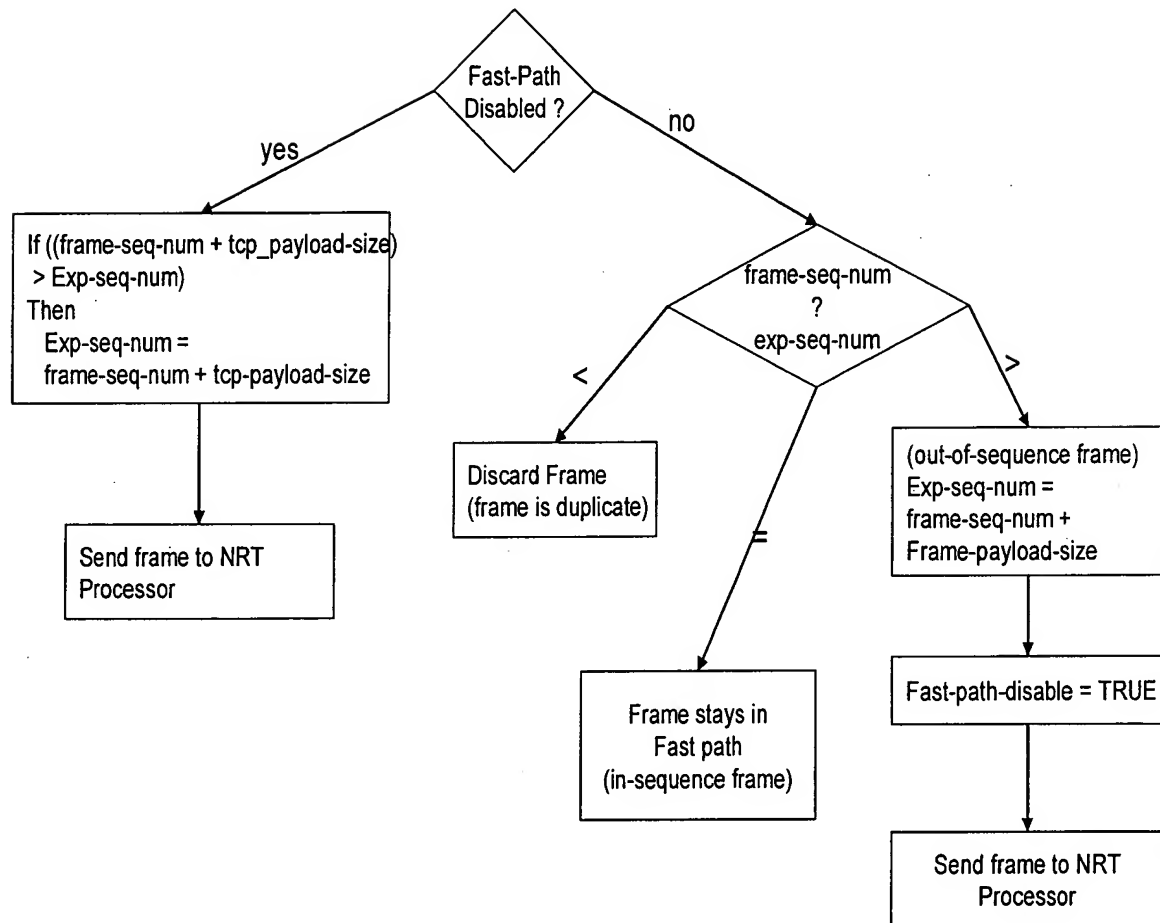


FIGURE 18A

SYSTEM AND METHODS FOR HIGH RATE HARDWARE-
ACCELERATED NETWORK PROTOCOL PROCESSING

Wang et al.

Appl. No.: Unknown Atty Docket: ISTOR.012A

byte#	#bits	field	Description
0	8	sp lo_pri ts_val rx_dis flush ddig_enb hdig_enb fp_enb	Control bits: 7: slow path forever (disallows automatic return to fast path) 6: low priority (0=high pri). If set and awr_prx_rdy_lo false, discard frame 5: TCP timestamp valid. If invalid, timestamp check automatically passes 4: PDU Rx disable. Do not store subsequent data or header segments 3: PDU flush. Do not store data segment, auto-reset flush at PDU end 2: data digest enable. Enables check of iSCSI data segment CRC 1: header digest enable. Enables check of iSCSI header segment CRC 0: fastpath enable. If disabled, entire frame stored to Rx chunk.
1-4	32	nxt_seq	Next TCP sequence number expected
5	8	ts	TCP timestamp [17:10]
6-9	32	pcrc	iSCSI partial digest (checked so far)
10-12	24 (1) (23)	dsctl wenb dsctr	iSCSI data segment control: 23: write SCSI data to buffer memory, vs header/non-SCSI data to Rx queue (flags how to interpret bytes 17-31) 22-0: data segment down counter (# words remaining in data seg, including data digest if present)
13-16	32	wptr	SCSI data write pointer ([3:0] indicates #qword residual bytes)
17-31 (17) (18) (19-21) (22) (23-26) (27-28) (29-31)	120 (8) (8) (24) (8) (32) (16) (24)	hctl state wres ahctr hoffst dplen dpoffst	Within iSCSI header segments: [7-4]=spare, [3]=final PDU, [2]=scsi data, [1:0]=#residual bytes [7-5]=spare, [4:0]= state Word residual (up to 3 bytes) Additional header segment down counter (#words in AHSs) Data offset from data out or data in header Data length from DPT Data offset from DPT, bits 31:24 (sb 512B boundary)
17-31	120	qres	Within SCSI data segments: Qword residual (up to 15 bytes)

FIGURE 18B

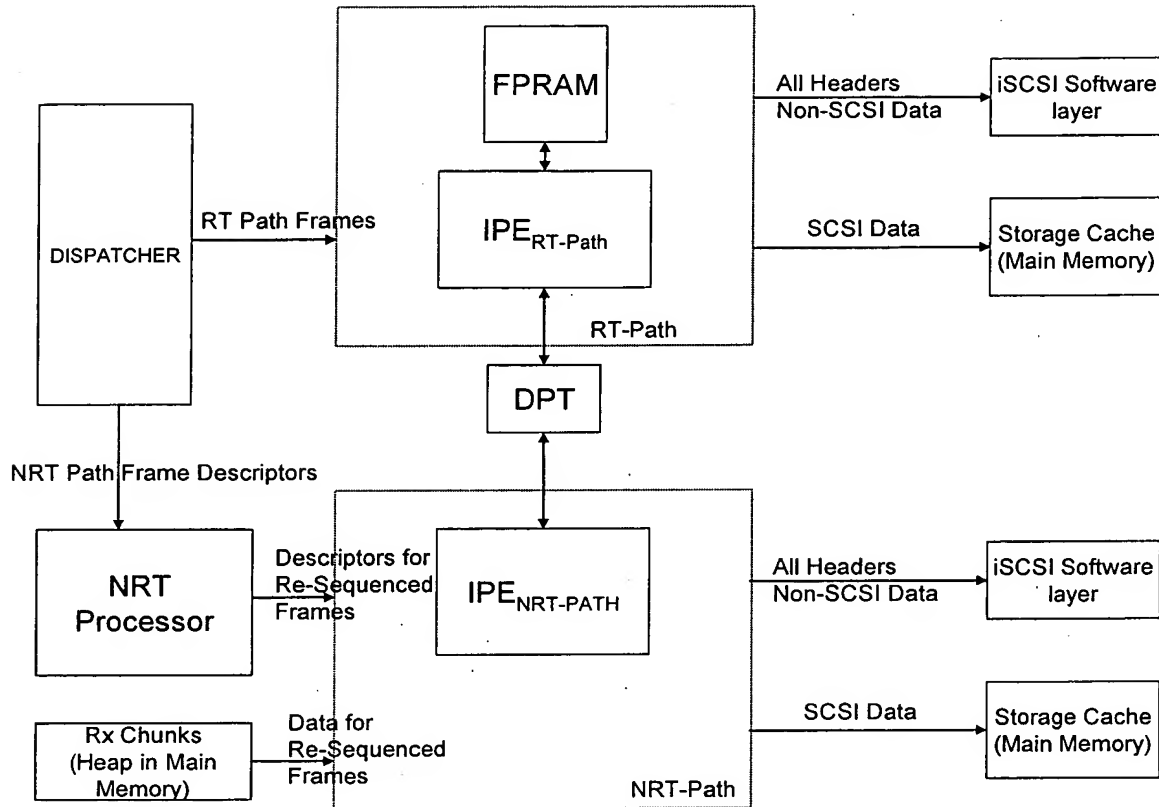


FIGURE 18C

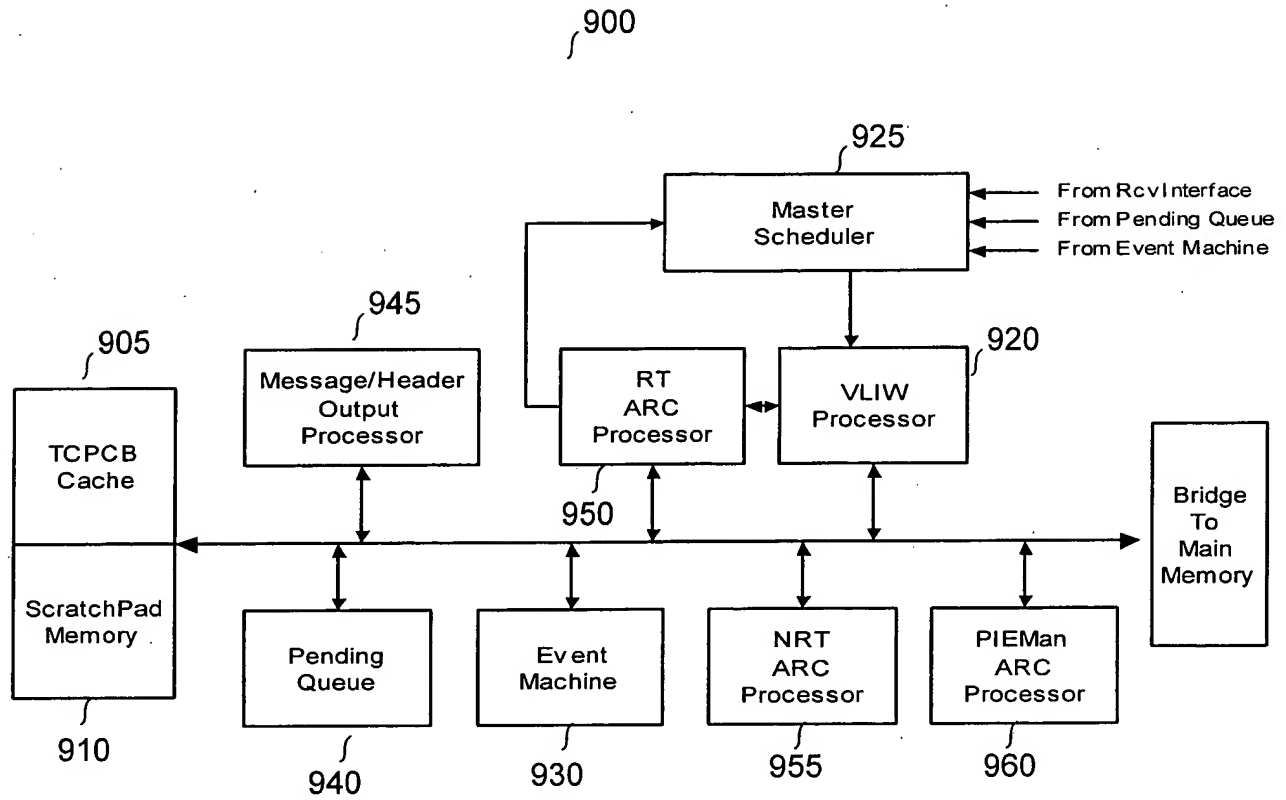


FIGURE 19

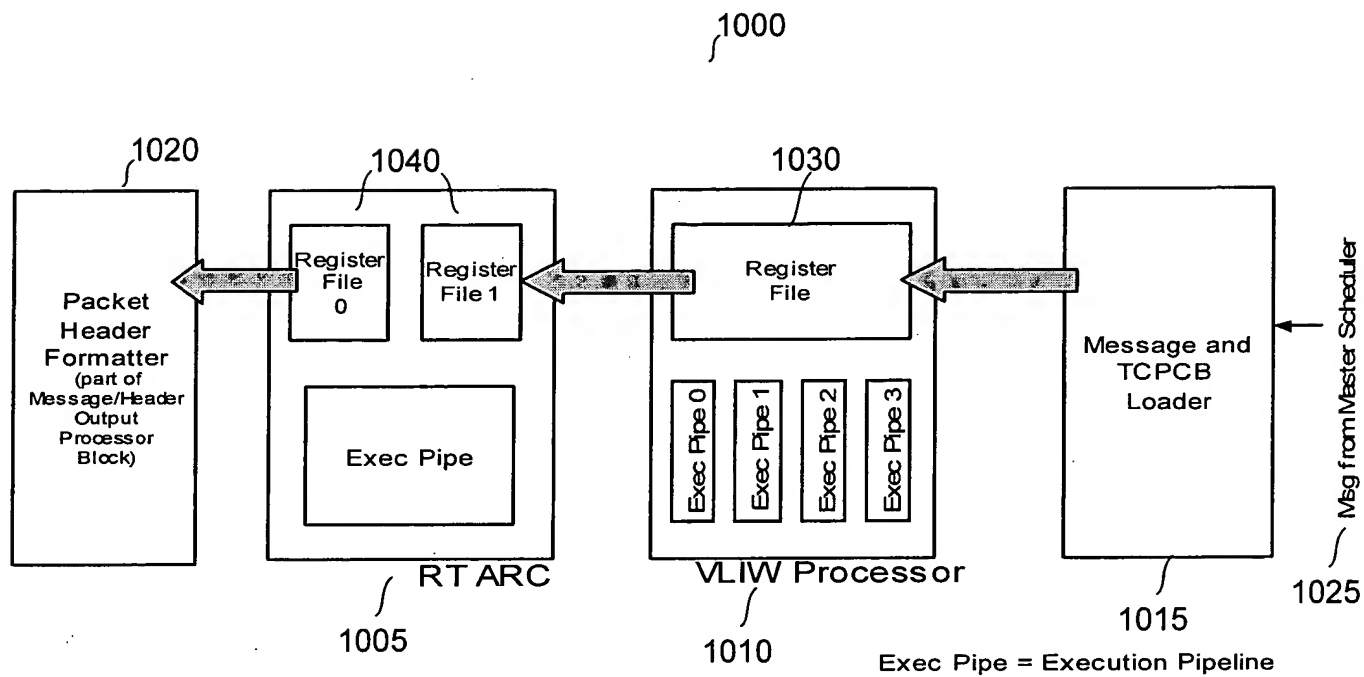


FIGURE 20

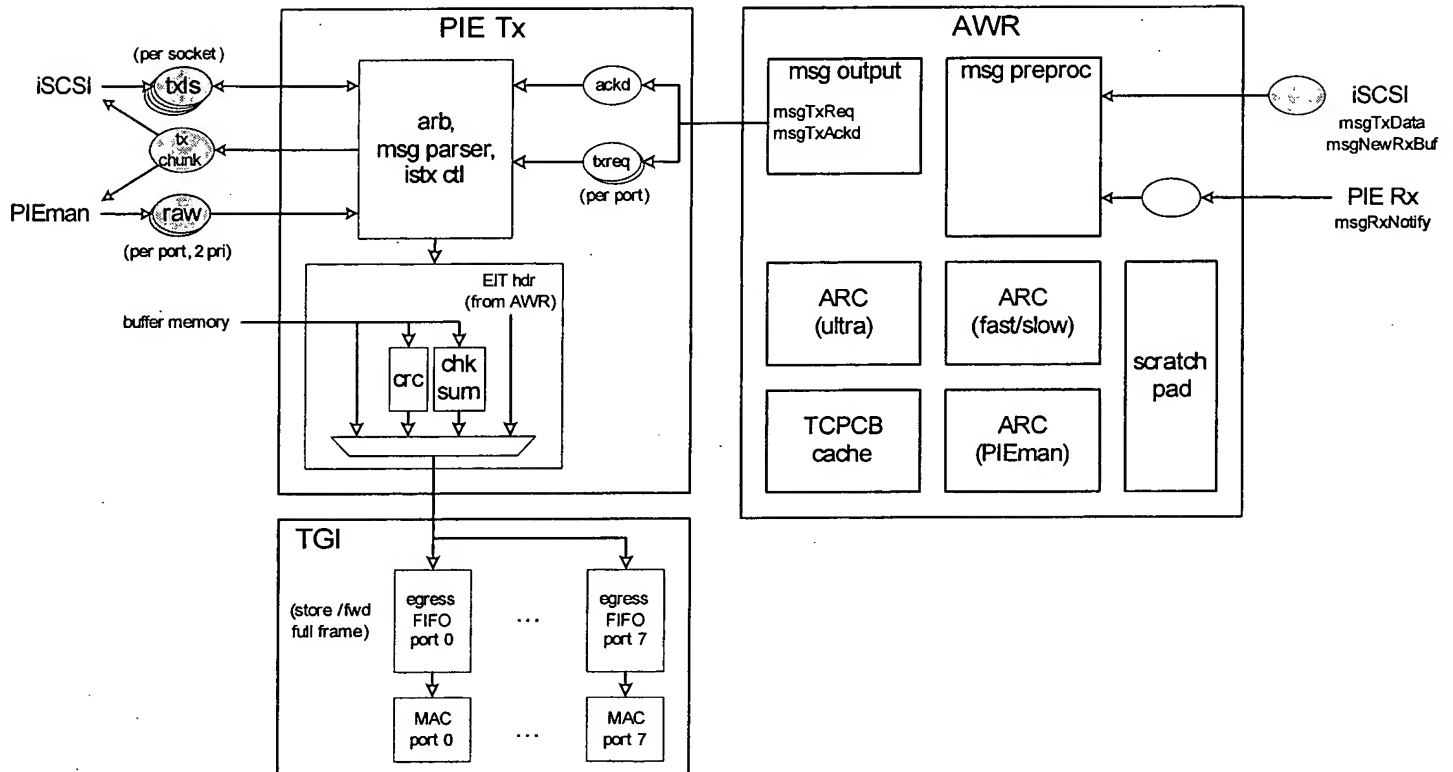


FIGURE 21

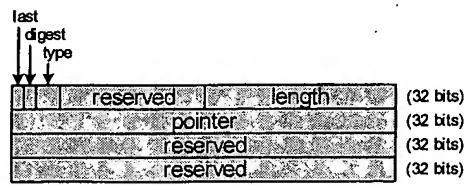


FIGURE 22

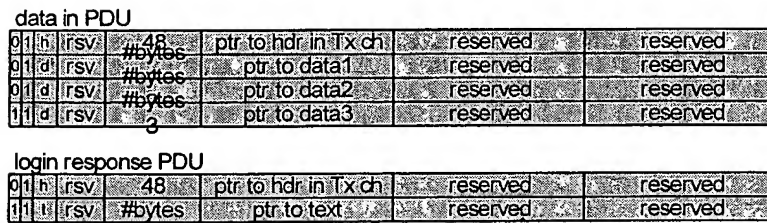


FIGURE 23

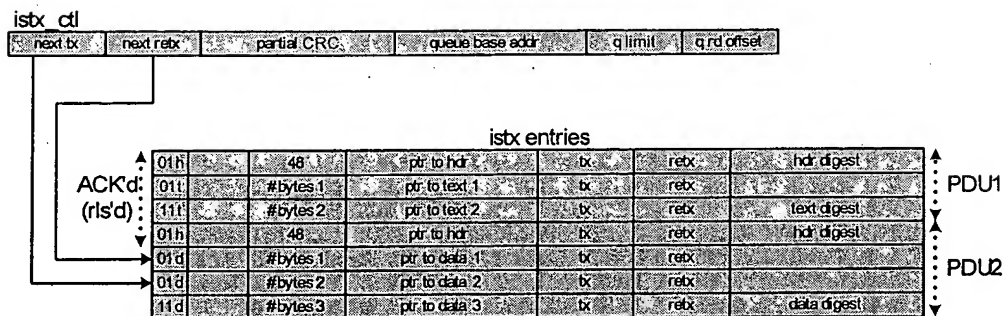


FIGURE 24



FIGURE 25

Command	Description
Push	Writes data beginning at write pointer and saves new write pointer in descriptor.
Push/Inc	Writes data beginning at write pointer, increments counter field by one, and saves new write pointer and counter in descriptor
Inc	Increments counter field by a specified amount and saves new counter in descriptor
Inc Bytes	Increments write pointer field by a specified amount and saves new write pointer in descriptor
Push/Chkpt	Writes data beginning at write pointer and saves new write pointer in descriptor and in descriptor extension as write checkpoint.
Push/Inc/Chkpt	Writes data beginning at write pointer, increments counter field by one, saves new write pointer and counter in descriptor, and saves new write pointer in descriptor extension as write checkpoint.
Inc/Chkpt	Increments counter field by a specified amount, saves new counter in descriptor, and copies current write pointer to write checkpoint
Rewind	Copies write checkpoint to write pointer and saves result in descriptor
Peek	Reads data beginning at read pointer (for queues) or write pointer (for stacks) but does not save new pointer in descriptor
Pop	Reads data beginning at read pointer (for queues) or write pointer (for stacks) and saves new pointer in descriptor
Pop/Dec	Reads data beginning at read pointer (for queues) or write pointer (for stacks), decrements counter field by one, and saves new pointer in descriptor
Dec	Decrement counter field by a specified amount and saves new counter in descriptor
Dec Bytes	Decrement read pointer (for queues) or write pointer (for stacks) by a specified amount and saves new pointer in descriptor

FIGURE 26

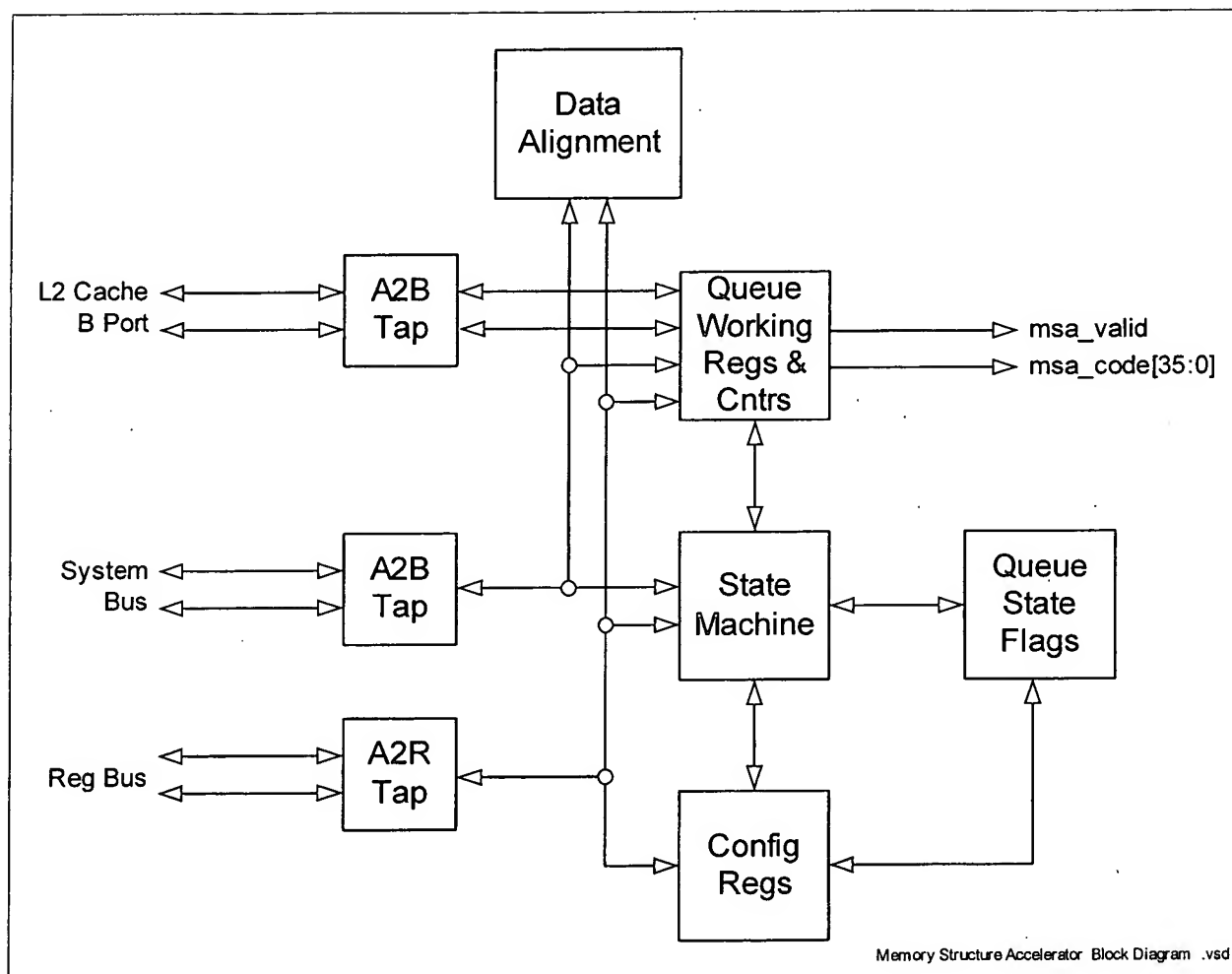


FIGURE 27

SYSTEM AND METHODS FOR HIGH RATE HARDWARE-
ACCELERATED NETWORK PROTOCOL PROCESSING

Wang et al.

Appl. No.: Unknown Atty Docket: ISTOR.012A

00000	---	01000	---	10000	---	11000	Rewind
00001	Pop/Dec (Read)	01001	Pop	10001		11001	Peek
00010	Push/Inc (Write)	01010	Push	10010	Push/Inc/Chkpt	11010	Push/Chkpt
00011	---	01011	Dec	10011	---	11011	---
00100	---	01100	---	10100	---	11100	---
00101	---	01101	---	10101	---	11101	---
00110	---	01110	Inc	10110	Inc/Chkpt	11110	---
00111	---	01111	---	10111	---	11111	---

FIGURE 28

00	Queue Not Empty	10	Queue Underflow
01	Queue Empty	11	Not used

FIGURE 29

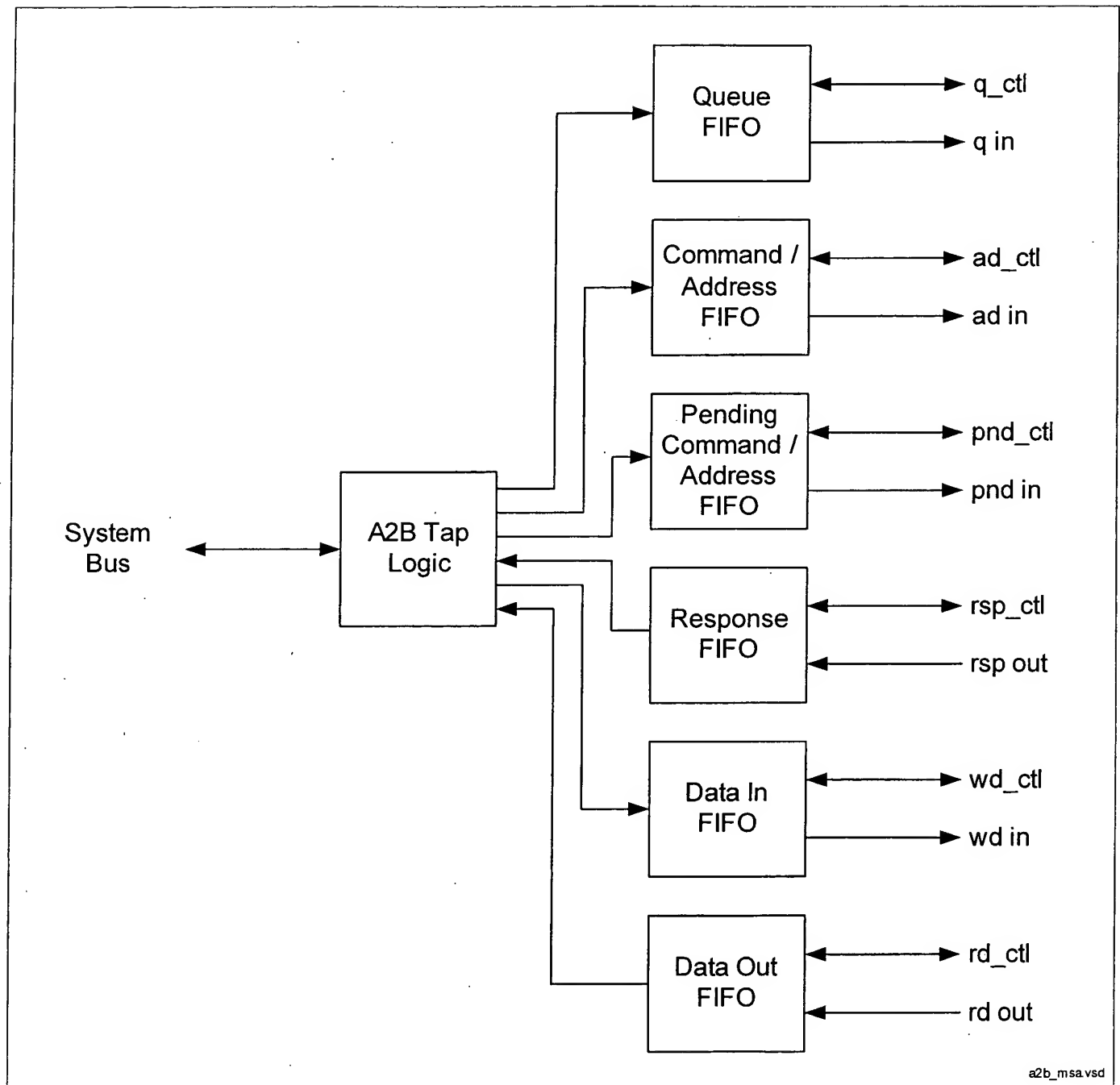


FIGURE 30

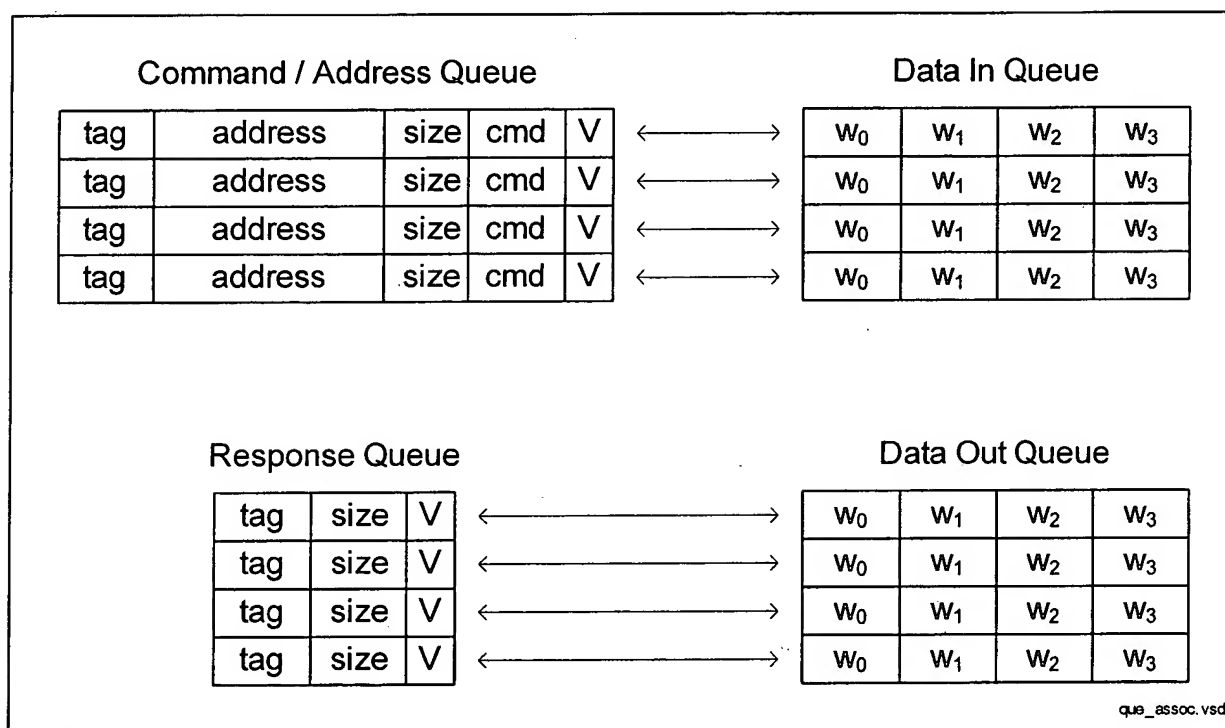
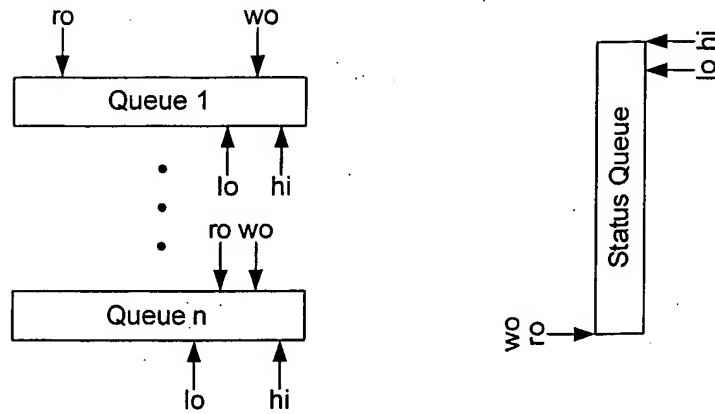
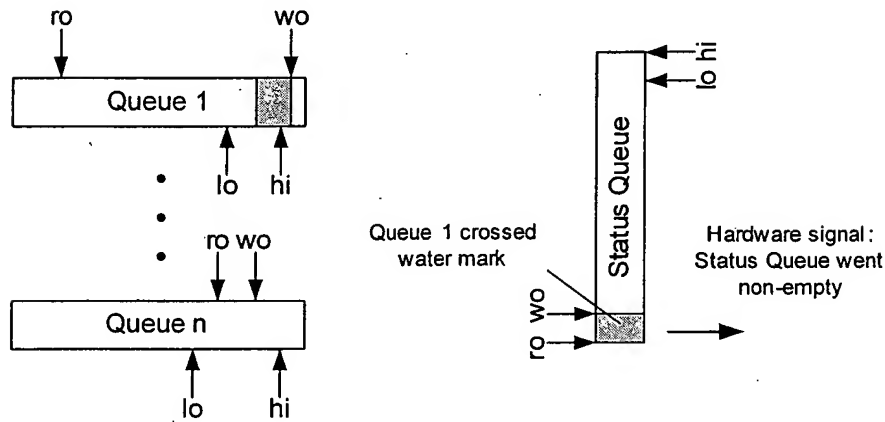


FIGURE 31

Current State



Push to Queue 1



Pop from Queue n

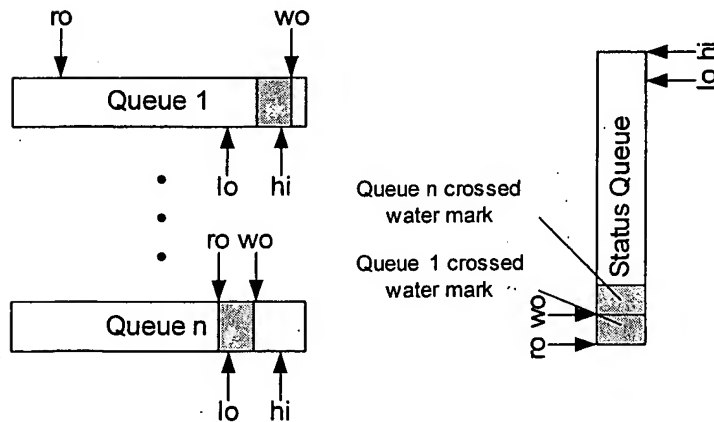


FIGURE 32

SYSTEM AND METHODS FOR HIGH RATE HARDWARE-
ACCELERATED NETWORK PROTOCOL PROCESSING

Wang et al.

Appl. No.: Unknown Atty Docket: ISTOR.012A

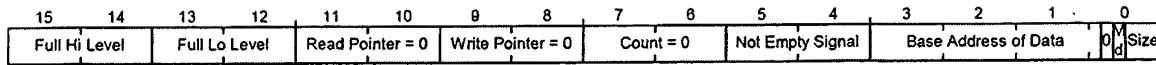


FIGURE 33

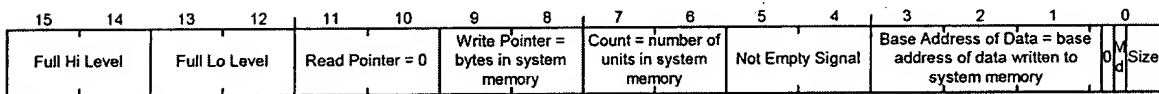


FIGURE 34